

# Curvas e Superfícies

M.C.F. de Oliveira

Fontes:

D.F. Rogers & J.A. Adams, Mathematical Elements for Computer Graphics, McGraw-Hill, 1999

Hearn & Baker, Cap. 8 (8-8 a 8-18)

An Interactive Introduction to Splines, on-line em  
<http://www.ibiblio.org/e-notes/Splines/Intro.htm>

# Modelagem Geométrica

- Disciplina que estuda como obter representações algébricas de curvas e superfícies que satisfazem determinadas propriedades
- Até o momento: assumimos objetos geométricos descritos por segmentos de reta ou polígonos (curvas/superfícies lineares por partes)
  - Em geral, descrevem aproximações de curvas e superfícies algébricas
  - Mas existem situações em que é necessário trabalhar com as representações exatas
    - mesmo quando utilizamos aproximações, p.ex., porque só podemos desenhar segmentos de reta e polígonos, é importante conhecer o objeto que está sendo aproximando

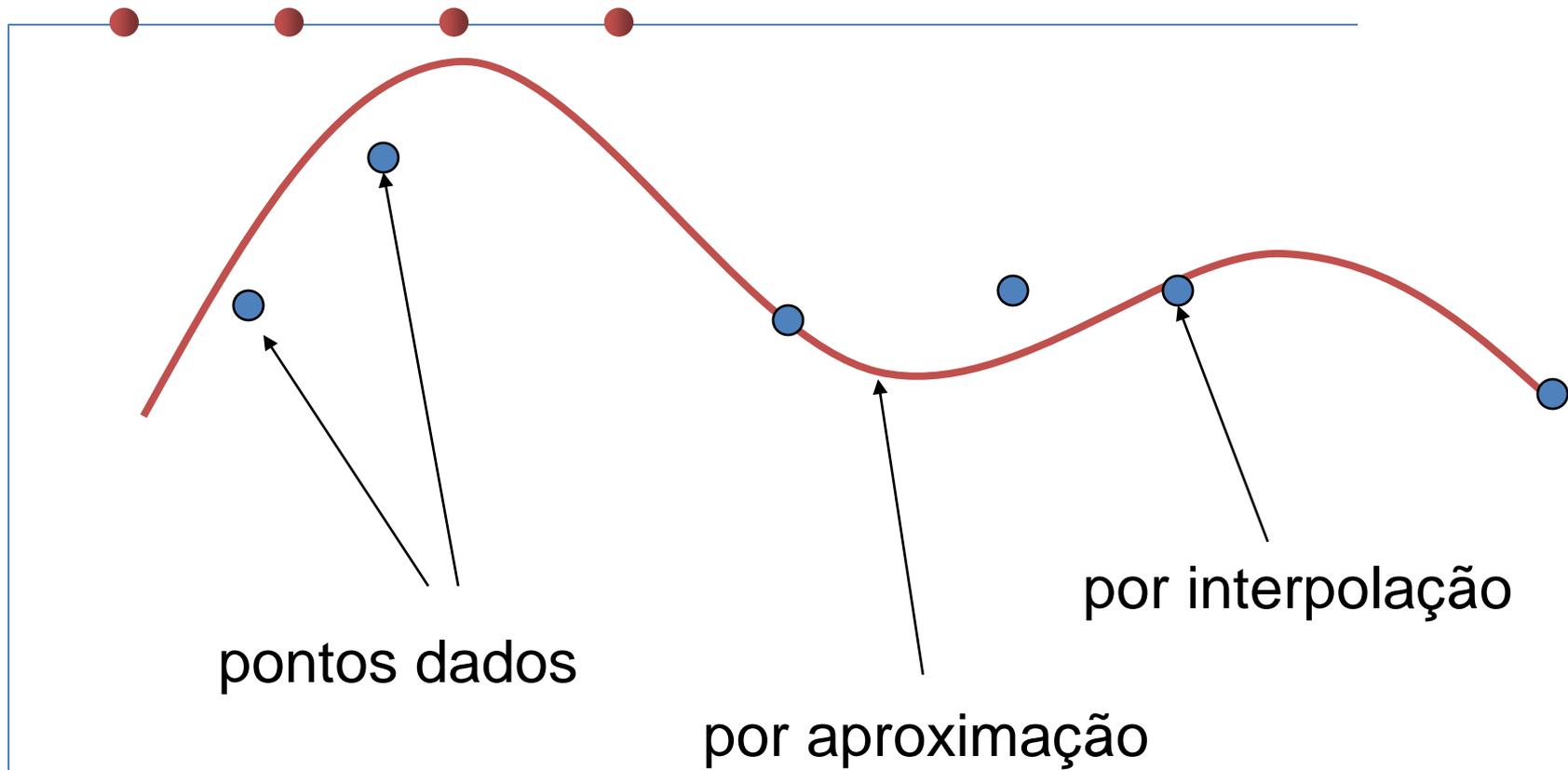
# Representação de Curvas

- Duas formas de representação
  - Conjunto de pontos que pertencem à curva
  - Analítica: formulação matemática
  - Vantagens
    - Precisão
    - Armazenagem compacta
    - Facilidade de cálculo (exato) de pontos intermediários
    - Facilidade para calcular propriedades como inclinação e curvatura
    - Facilidade para desenhar as curvas
    - Facilidade para fazer alterações contínuas no formato da curva (*design*)

# Ajuste x Aproximação

- Dado um conjunto de pontos, obter uma representação analítica para uma curva que os aproxima
- Ajuste de curvas (*curve fitting*)
  - Uma curva que ajusta (*fit*) os pontos dados passa por todos esses pontos (interpolação)
  - Técnica usual: *splines* cúbicas (aproximação polinomial por partes)
- Aproximação de curvas (*curve fairing*)
  - Uma curva que aproxima (*fair*) os pontos dados pode não passar por nenhum deles, mas mostra a tendência dos dados.
  - Ex. pontos coletados ou obtidos em medidas experimentais

# Modelagem de curvas



# Representação Implícita x Explícita

- Representação explícita:  $y = f(x)$ 
  - Ex.:  $y = mx + b$
  - Uma única equação não representa curvas fechadas, ou com múltiplos valores de  $y$  para um dado  $x$
- Representação implícita:  $f(x, y) = 0$ 
  - Ex.: equação implícita de 2o. grau genérica engloba uma variedade de curvas bidimensionais denominadas seções cônicas
  - parábola, hipérbole, elipse, círculo, ...
- Ambas são representações não paramétricas

# Representação não-paramétrica

## ■ Limitações

- Inadequada para representar curvas fechadas, ou com múltiplos valores de  $y$
- Dependentes do sistema de coordenadas, cuja escolha afeta a facilidade de uso
- Pontos em uma curva calculados a partir de incrementos uniformes em  $x$  não estão distribuídos uniformemente ao longo da curva
  - Qualidade de traçado fica prejudicada

# Representação paramétrica

- As coordenadas de pontos na curva são representadas como uma função de um único parâmetro: a posição do ponto na curva é fixada pelo valor do parâmetro
  - Ex. para uma curva 2D que usa  $t$  como parâmetro, as coordenadas cartesianas de um ponto na curva são dadas por:  
 $x = x(t); y = y(t)$
  - Vetor posição de um ponto:  $\mathbf{P}(t) = [x(t) \ y(t)]$
  - Derivada em P (vetor tangente à curva):  $\mathbf{P}'(t) = [x'(t) \ y'(t)]$
  - Inclinação:  $dy/dx = (dy/dt)/(dx/dt) = y'(t)/x'(t)$

# Representação paramétrica

- Adequada para representar curvas fechadas e com múltiplos valores de  $y$  para um dado  $x$
- Forma não-paramétrica pode ser obtida eliminando-se o parâmetro
  - dado  $x$ , para determinar  $y$  basta obter o valor do parâmetro  $t$ , a partir de  $x$ , e usar esse valor para obter  $y$
- Independente do sistema de coordenadas
- Ambas as formas, paramétrica e não paramétrica, têm vantagens e desvantagens em situações específicas!

# Exemplo

- Dados 2 vetores que especificam posições iniciais  $P_1$  e  $P_2$ , possível representação paramétrica do segmento de reta:

$$P(t) = P_1 + (P_2 - P_1)t, \quad 0 \leq t \leq 1$$

- Como  $P(t)$  é vetor de posição, cada um de seus componentes têm uma representação paramétrica  $x(t)$  e  $y(t)$  entre  $P_1$  e  $P_2$ , i.e.,

$$x(t) = x_1 + (x_2 - x_1)t, \quad 0 \leq t \leq 1$$

$$y(t) = y_1 + (y_2 - y_1)t, \quad 0 \leq t \leq 1$$

- Ex.

# Exemplo

- Círculo no primeiro quadrante
  - Representação paramétrica não é única!
  - Ex.

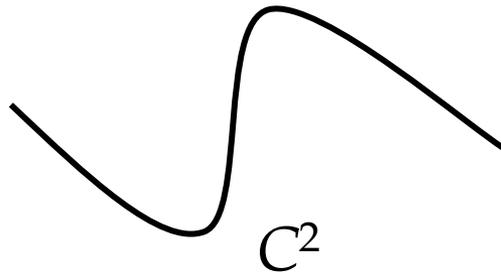
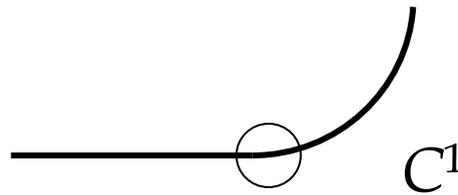
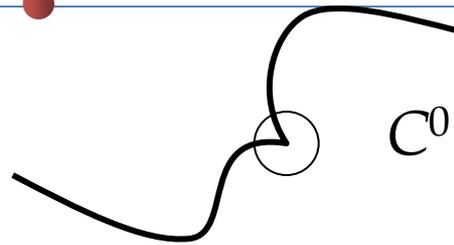
# Curvas e Superfícies Paramétricas

- A forma paramétrica é comum na modelagem geométrica
  - Permite que a curva/superfície seja desenhada (aproximada) facilmente
  - Permite indicar que trechos da curva/superfície serão usados
  - Manipulação algébrica mais simples
- Curva em 3D é dada por
  - $C(t) = [C_x(t) \ C_y(t) \ C_z(t)]^T$
- Superfície em 3D é dada por
  - $S(u, v) = [S_x(u, v) \ S_y(u, v) \ S_z(u, v)]^T$

# Continuidade

- Normalmente queremos curvas e superfícies “suaves”
- Critério de “suavidade” associado com critério de continuidade algébrica
  - Continuidade  $C^0$  → funções paramétricas são contínuas, isto é, sem “pulos”
  - Continuidade  $C^1$  → funções paramétricas têm primeiras derivadas contínuas, isto é, tangentes variam suavemente
  - Continuidade  $C^k$  → funções paramétricas têm  $k$ -ésimas derivadas contínuas

# Continuidade



# Curvas de Bézier

- Técnicas de aproximação de curvas são muito usadas em ambientes de projeto (CAD) interativos, por serem mais intuitivas do que técnicas de ajuste
- Método adequado para o *design* de curvas e superfícies de forma livre em ambientes interativos foi desenvolvido por Pierre Bézier
- Uma curva de Bézier é determinada por um conjunto de pontos de controle (polígono de controle)
- Curvas podem ser lineares, quadráticas, cúbicas, etc., dependendo do número de pontos de controle e da ordem do polinômio usado para obter a aproximação

# Algoritmo de De Casteljaou

- Suponha que desejamos aproximar uma curva polinomial entre dois pontos dados,  $\mathbf{p}_0$  e  $\mathbf{p}_1$
- A solução natural é um segmento de reta que passa por  $\mathbf{p}_0$  e  $\mathbf{p}_1$  cuja parametrização mais comum é
$$\mathbf{p}(u) = (1 - u) \mathbf{p}_0 + u \mathbf{p}_1$$
- Podemos pensar em  $\mathbf{p}(u)$  como uma média ponderada entre  $\mathbf{p}_0$  e  $\mathbf{p}_1$
- Note que os polinômios  $(1 - u)$  e  $u$  somam 1 para qualquer valor de  $u$ 
  - São chamados de funções de mistura (*blending functions*)



# Algoritmo de De Casteljau

- Para generalizar a idéia para três pontos  $\mathbf{p}_0$ ,  $\mathbf{p}_1$  e  $\mathbf{p}_2$  consideramos primeiramente os segmentos de reta  $\mathbf{p}_0\mathbf{p}_1$  e  $\mathbf{p}_1\mathbf{p}_2$

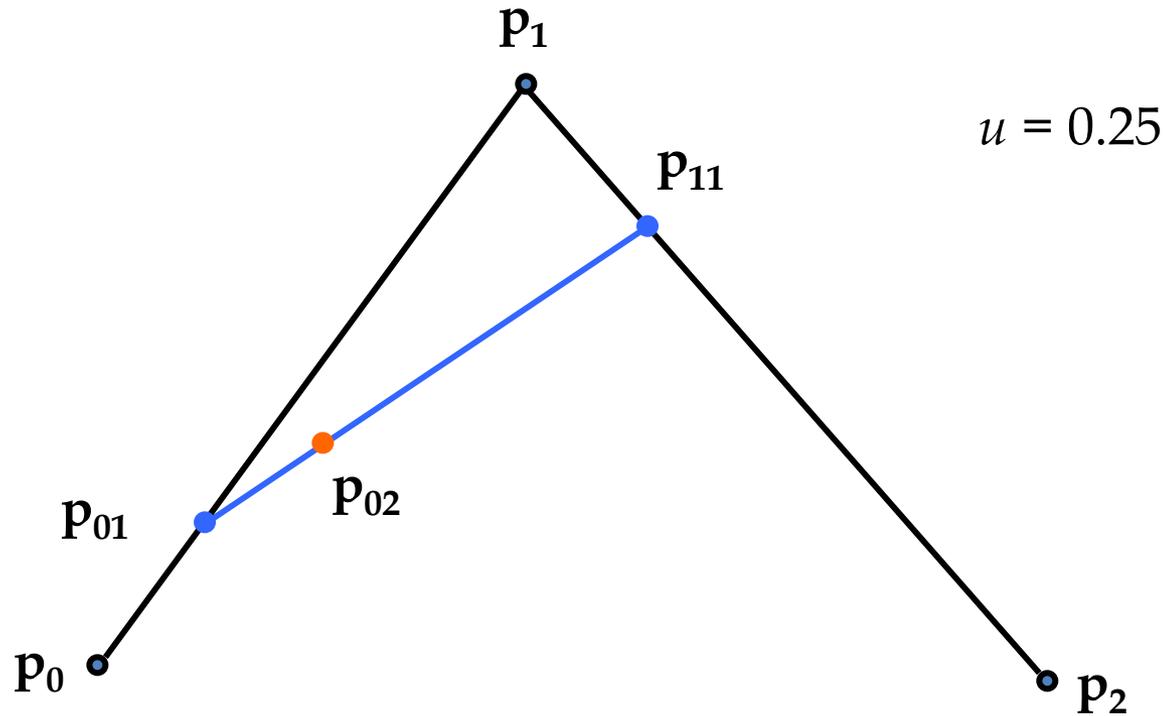
$$\mathbf{p}_{01}(u) = (1 - u) \mathbf{p}_0 + u \mathbf{p}_1$$

$$\mathbf{p}_{11}(u) = (1 - u) \mathbf{p}_1 + u \mathbf{p}_2$$

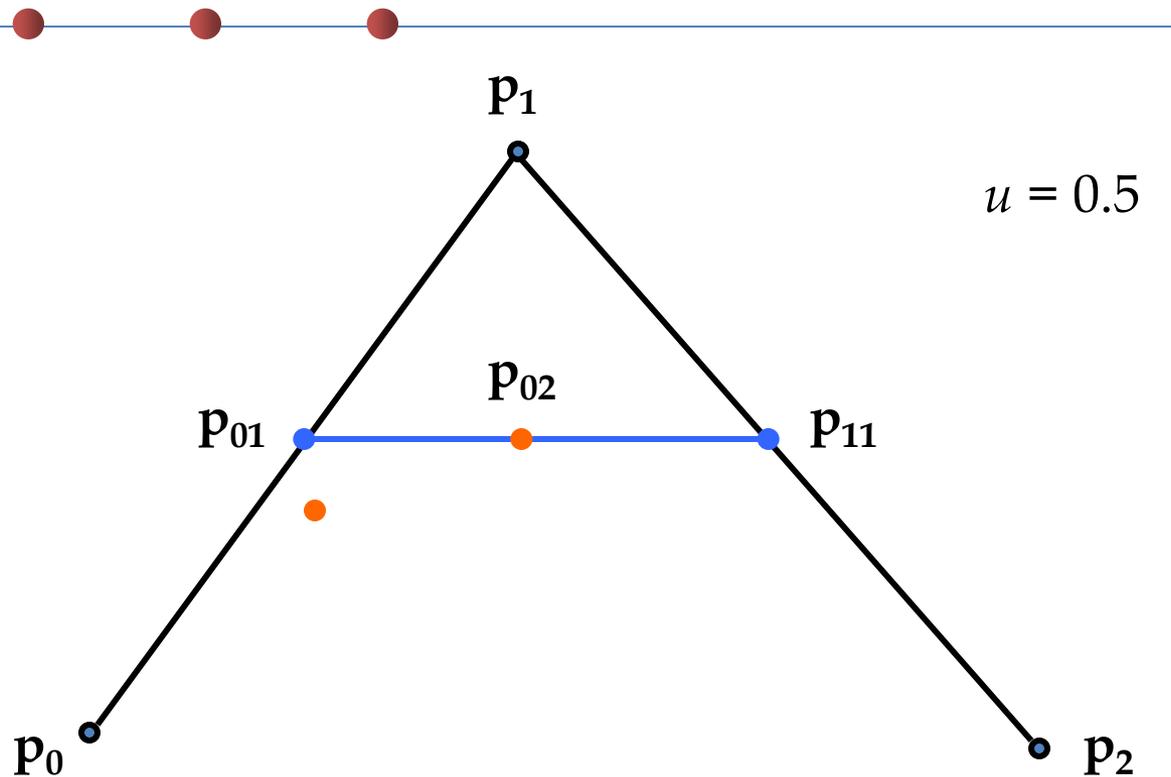
- Podemos agora realizar uma interpolação entre  $\mathbf{p}_{01}(u)$  e  $\mathbf{p}_{11}(u)$

$$\begin{aligned} \mathbf{p}_{02}(u) &= (1 - u) \mathbf{p}_{01}(u) + u \mathbf{p}_{11}(u) \\ &= (1 - u)^2 \mathbf{p}_0 + 2u(1 - u) \mathbf{p}_1 + u^2 \mathbf{p}_2 \end{aligned}$$

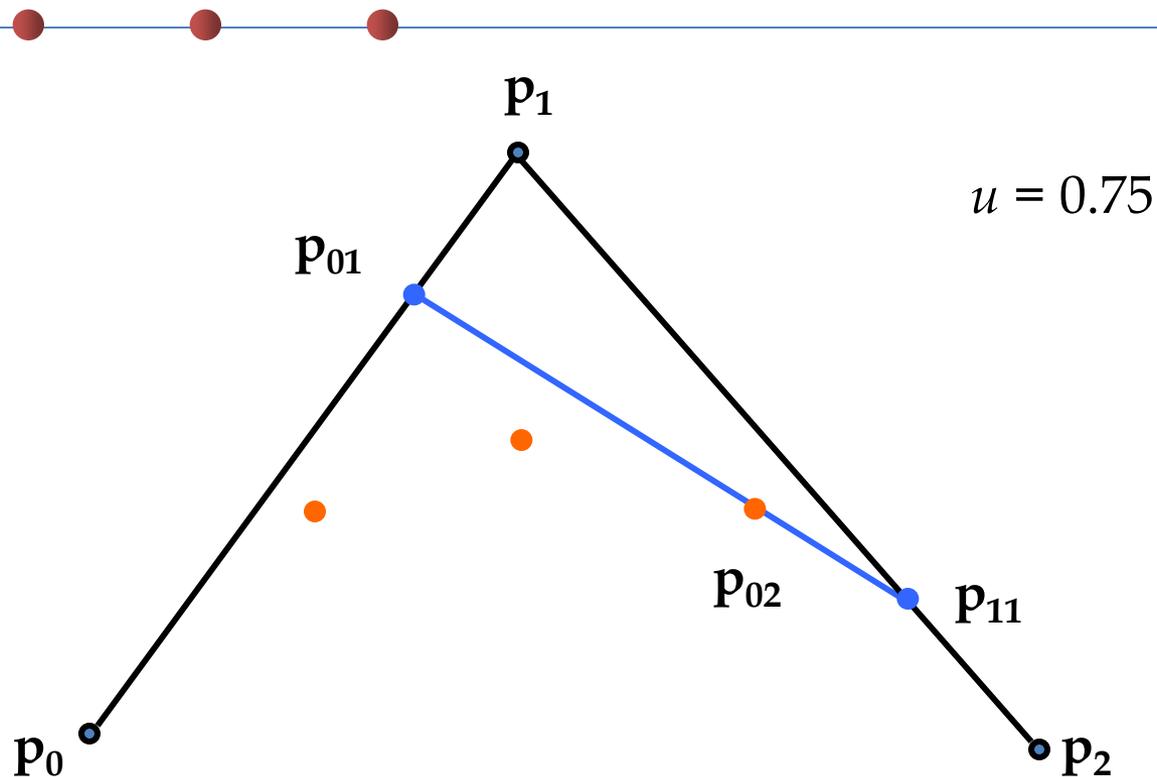
# Algoritmo de De Casteljau



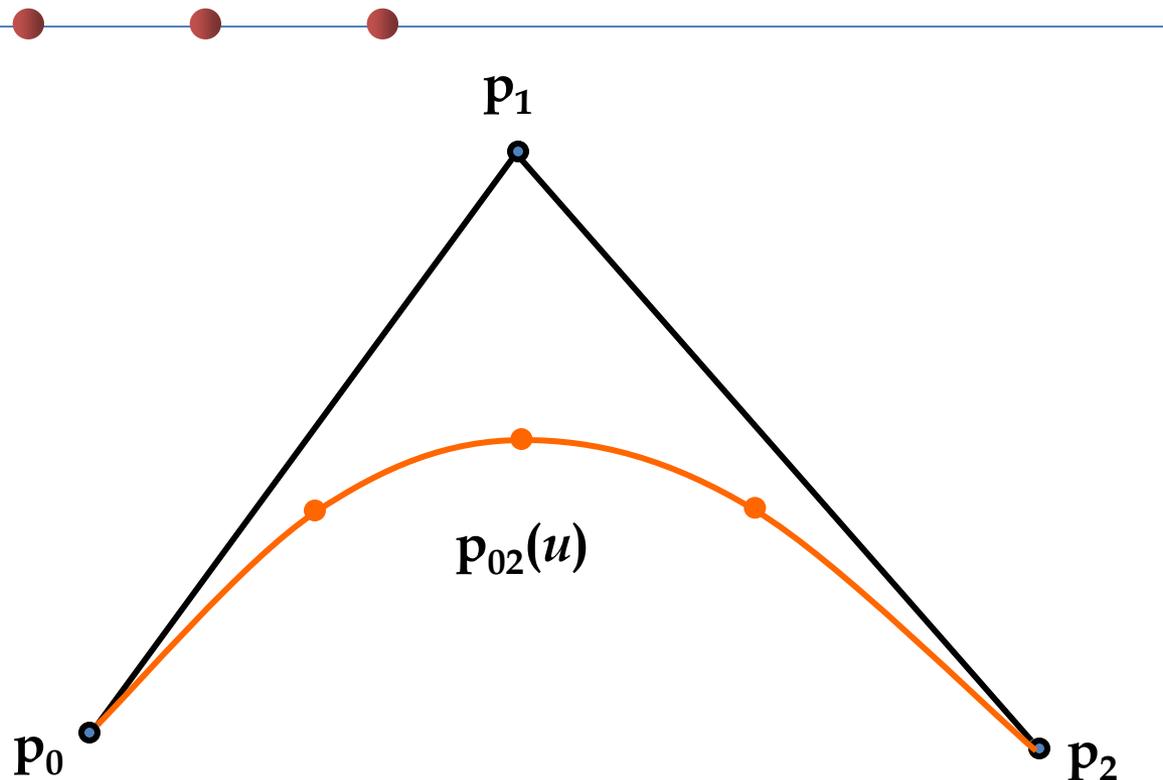
# Algoritmo de De Casteljau



# Algoritmo de De Casteljau



# Algoritmo de De Casteljau



# Algoritmo de De Casteljaou

- A curva obtida pode ser entendida como a “mistura” dos pontos  $\mathbf{p}_0$ ,  $\mathbf{p}_1$  e  $\mathbf{p}_2$  por intermédio de três funções quadráticas:

- $b_{02}(u) = (1 - u)^2$
- $b_{12}(u) = 2u(1 - u)$
- $b_{22}(u) = u^2$

- Aplicando mais uma vez a idéia podemos definir uma cúbica por 4 pontos

$$\mathbf{p}_{02}(u) = (1 - u)^2 \mathbf{p}_0 + 2u(1 - u) \mathbf{p}_1 + u^2 \mathbf{p}_2$$

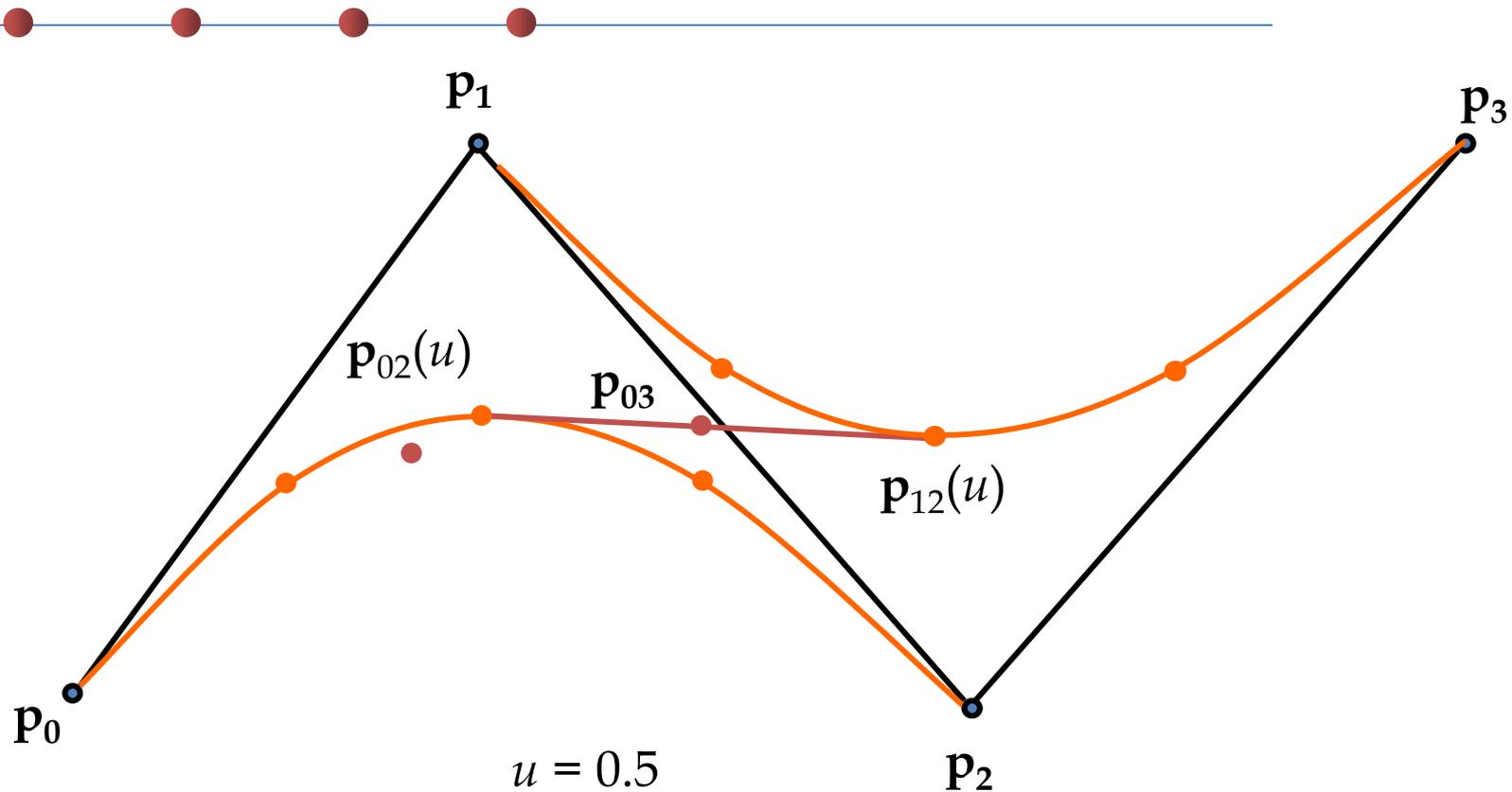
$$\mathbf{p}_{12}(u) = (1 - u)^2 \mathbf{p}_1 + 2u(1 - u) \mathbf{p}_2 + u^2 \mathbf{p}_3$$

$$\mathbf{p}_{03}(u) = (1 - u) \mathbf{p}_{02}(u) + u \mathbf{p}_{12}(u)$$

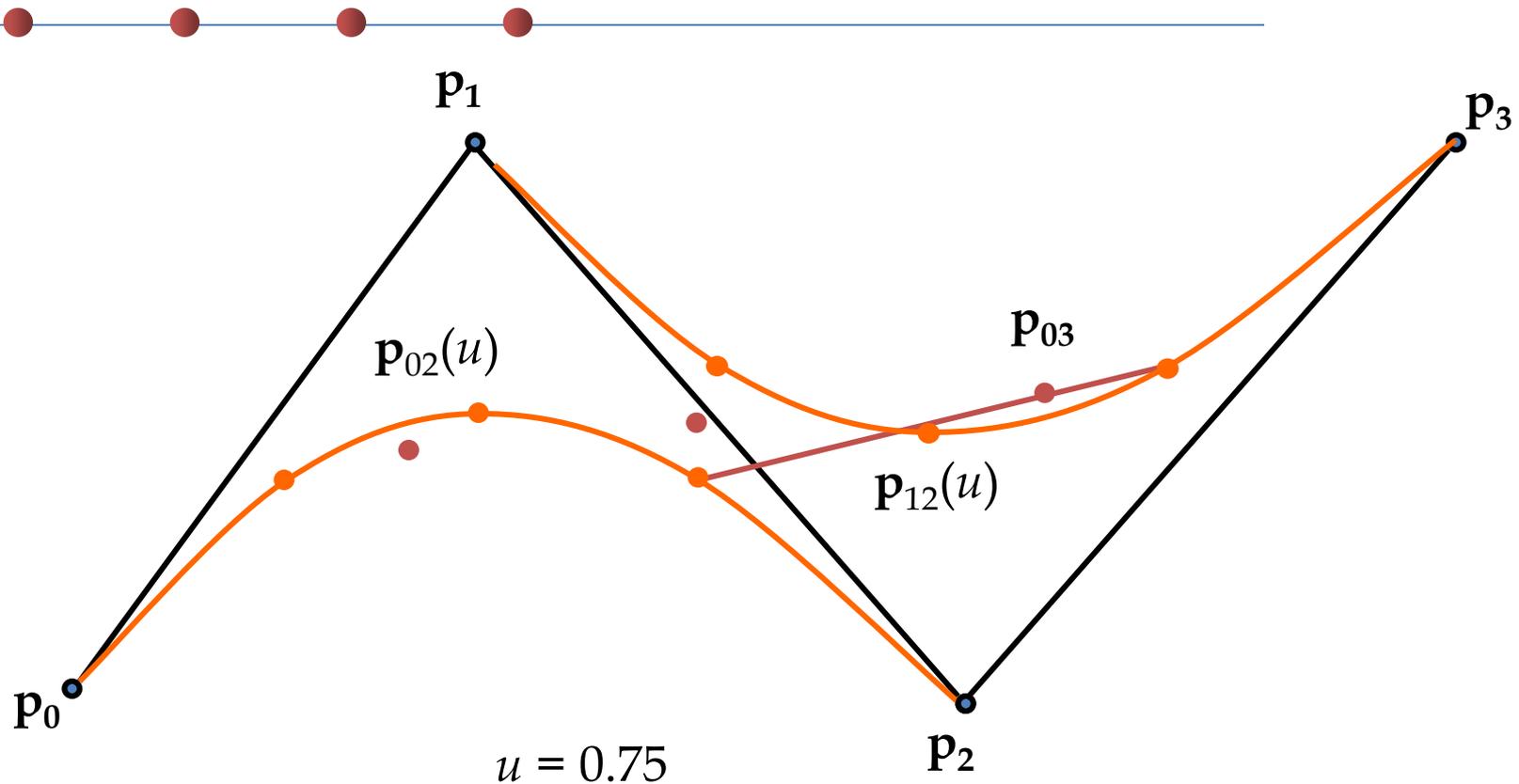
$$= (1 - u)^3 \mathbf{p}_0 + 3u(1 - u)^2 \mathbf{p}_1 + 3u^2(1 - u) \mathbf{p}_2 + u^3 \mathbf{p}_3$$



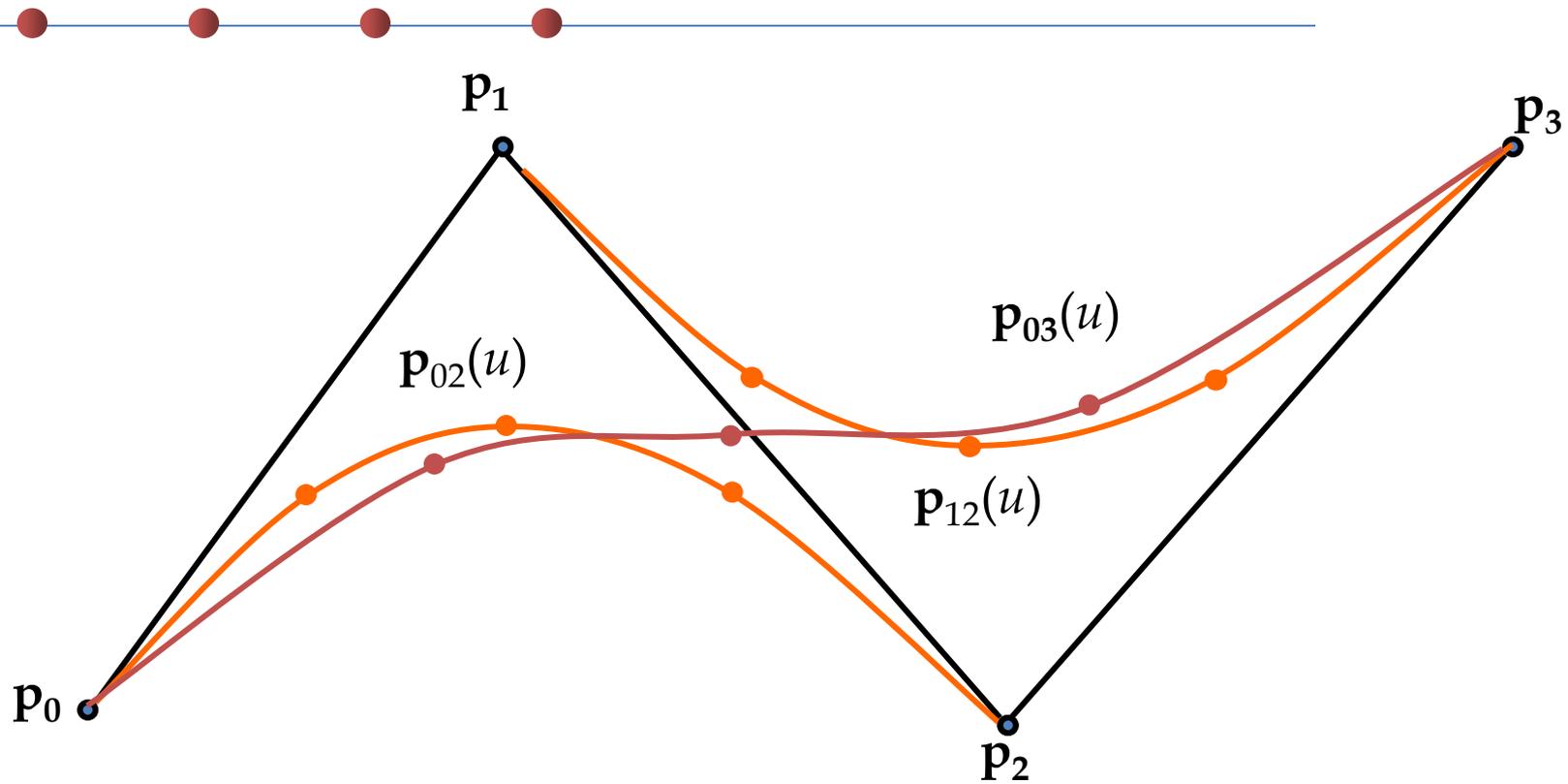
# Algoritmo de De Casteljau



# Algoritmo de De Casteljau



# Algoritmo de De Casteljau



# Algoritmo de De Casteljau

- Novamente temos uma curva dada pela soma de 4 funções de mistura (agora cúbicas), cada uma multiplicada por um dos 4 pontos
  - $b_{03}(u) = (1 - u)^3$
  - $b_{13}(u) = 3 u (1 - u)^2$
  - $b_{23}(u) = 3 u^2 (1 - u)$
  - $b_{33}(u) = u^3$
- Em geral, uma curva de grau  $n$  pode ser construída desta forma e será expressa por

$$\mathbf{p}_{0n}(u) = \sum_{j=0}^n b_{jn}(u) \mathbf{p}_j$$

# Curvas de Bézier e Polinômios de Bernstein

- As curvas construídas pelo algoritmo de De Casteljau são conhecidas como *curvas de Bézier* e as funções de mistura são chamadas de *base Bézier* ou *polinômios de Bernstein*
- Observamos que os polinômios de Bernstein de grau  $n$  têm como forma geral  $b_{i,n}(u) = c_i u^i (1 - u)^{n-i}$
- Se escrevermos as constantes  $c_i$  para os diversos polinômios, teremos
  - 1º grau: 1 1
  - 2º grau: 1 2 1
  - 3º grau: 1 3 3 1
  - 4º grau: 1 4 6 4 1
- Vemos que o padrão de formação corresponde ao *Triângulo de Pascal* e portanto, podemos escrever

$$b_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}$$

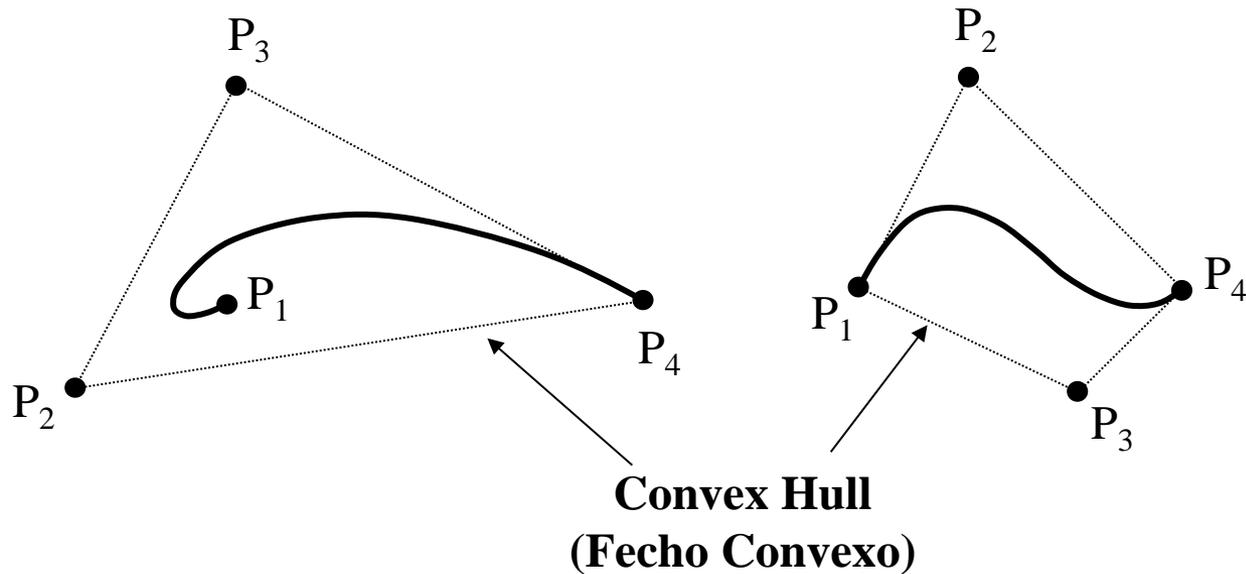
28

# Curvas de Bézier

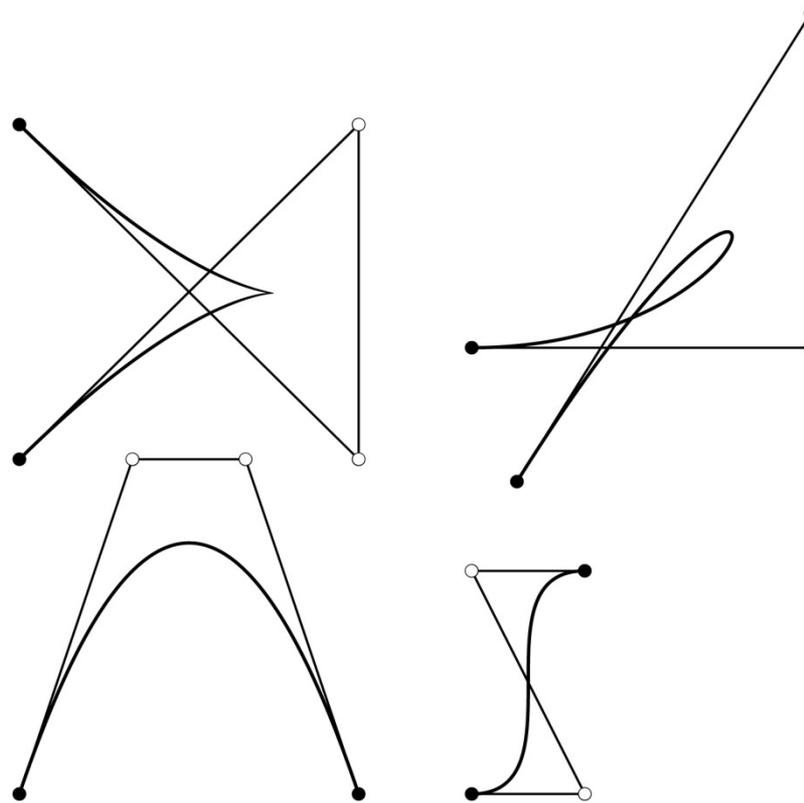
- Ver <http://www.ibiblio.org/e-notes/Splines/Bezier.htm>
- Por construção, as curvas de Bézier passam pelos pontos de controle terminais, i.e.  
$$\mathbf{P}(0) = \mathbf{P}_0, \quad \mathbf{P}(1) = \mathbf{P}_2$$
- Matematicamente, uma curva de Bézier paramétrica é definida como  
$$\mathbf{P}(t) = \sum_{i=0,n} B_i^n(t) \mathbf{P}_i \quad (*)$$
- $B_i^n(t)$  são as funções base de Bernstein de ordem  $n$
- $n$ , o grau das funções base, é igual ao número de pontos do polígono de controle menos 1

# Curvas de Bézier

Curvas para pontos de controle,  $P_1$ ,  $P_2$ ,  $P_3$ , e  $P_4$



# Curvas de Bézier



# Curvas de Bézier

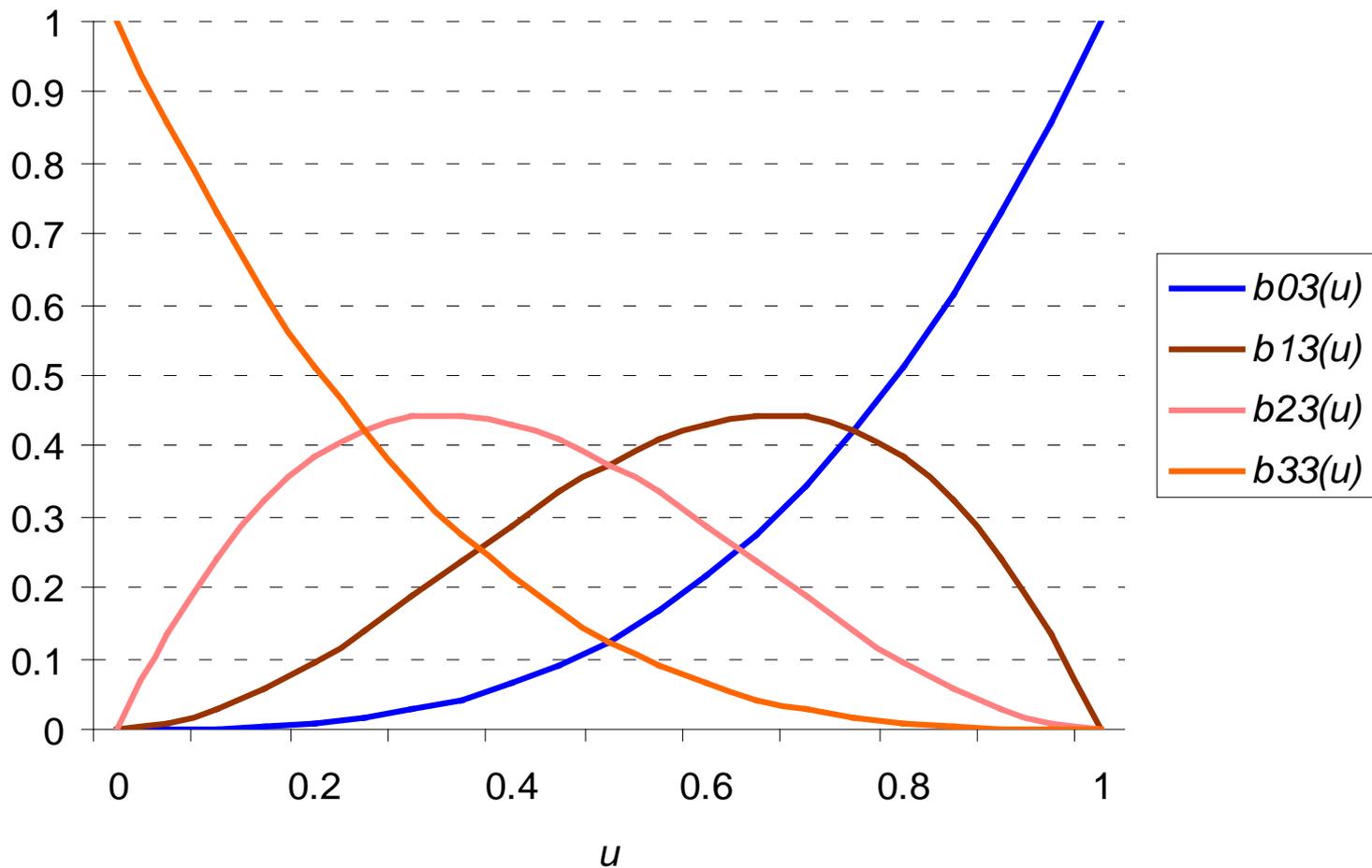
- Para 'desenhar' a curva que aproxima  $n+1$  pontos dados, pode-se usar o algoritmo iterativo de DeCasteljau:

$$\mathbf{P}_i^j(t) = (1-t)\mathbf{P}_i^{j-1} + t\mathbf{P}_{i+1}^{j-1}, \quad j = 1, n \quad i = 0, n-j$$

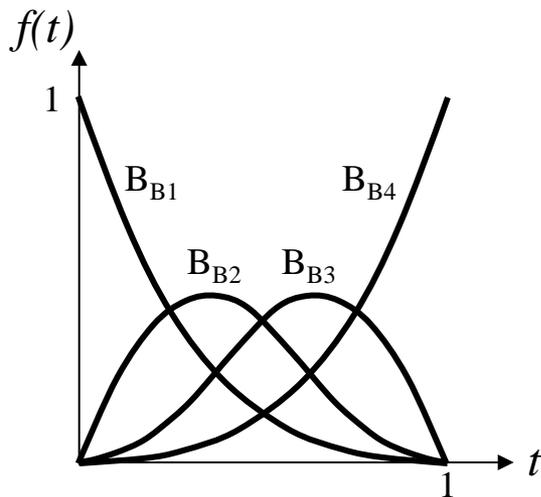
- Exemplo para  $n = 3$  em <http://www.ibiblio.org/e-notes/Splines/Bezier.htm>
- Alternativamente, pode-se usar a Eq. (\*)

$$\mathbf{P}(t) = \sum_{i=0,n} B_i^n(t) \mathbf{P}_i, \quad (*)$$
$$B_i^n(t) = C_n^i (1-t)^{n-i} t^i, \quad C_n^i = n! / i!(n-i)!$$

## Polinômios de Bernstein de grau 3



# Funções de Blending de Bézier



**Polinômios de Bernstein:**

$$B_{B_1} = (1-t)^3 \quad B_{B_2} = 3t(1-t)^2$$

$$B_{B_3} = 3t^2(1-t) \quad B_{B_4} = t^3$$

$$C(t) = (-t^3 + 3t^2 - 3t + 1)P_1 + \\ (3t^3 - 6t^2 + 3t)P_2 + \\ (-3t^3 + 3t^2)P_3 + \\ t^3 P_4$$

# Forma Matricial da Base Bézier

- Podemos escrever a equação para uma curva de Bézier cúbica na forma

$$\mathbf{p}(u) = \mathbf{p}_{03}(u) = [1 \quad u \quad u^2 \quad u^3] \mathbf{M}_B \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

onde  $\mathbf{M}_B$  é a matriz de coeficientes da base Bézier

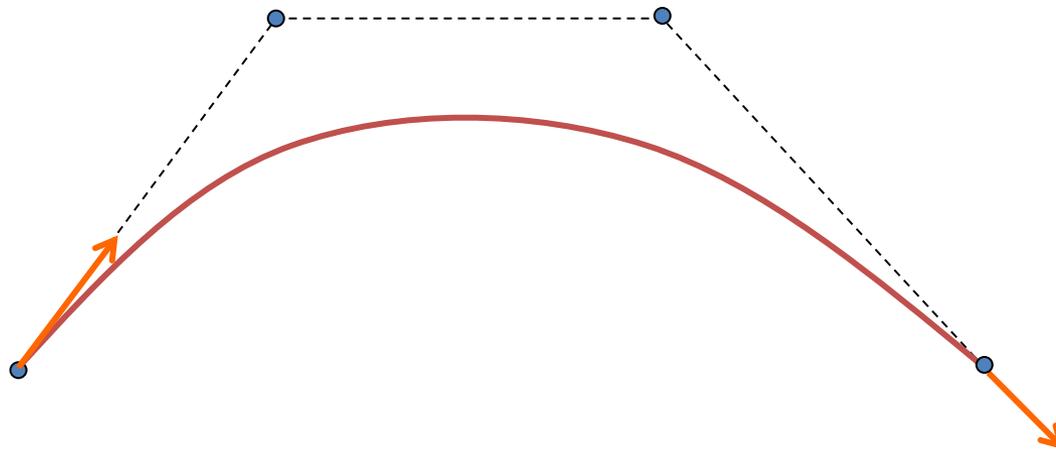
$$\mathbf{M}_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

# Curvas de Bézier - propriedades

- Curva interpola o primeiro e último pontos do polígono de controle
- Forma da curva 'acompanha' a forma do polígono de definição  $\Rightarrow$  formulação adequada ao '*design*' interativo de formas
- Curva está contida no fecho convexo do polígono de definição
  - Os polinômios de Bernstein somam 1 para qualquer  $u$
- Continuidade infinita: todas as derivadas são contínuas
- Qualquer linha reta intercepta a curva tantas ou menos vezes quanto intercepta o polígono de controle
  - Não pode oscilar demasiadamente

# Curvas de Bézier - propriedades

- A curva é invariante sob transformações geométricas afins (rotação, translação, escala, ...)
  - Transformar os pontos de controle e desenhar a curva é equivalente a desenhar a curva transformada
- As tangentes à curva em  $\mathbf{p}_0$  e  $\mathbf{p}_n$  têm a direção dos segmentos de reta  $\mathbf{p}_0\mathbf{p}_1$  e  $\mathbf{p}_{n-1}\mathbf{p}_n$ , respectivamente
  - Para cúbicas, as derivadas são  $3(\mathbf{p}_1 - \mathbf{p}_0)$  e  $3(\mathbf{p}_2 - \mathbf{p}_3)$



# Curvas de Bézier - Exemplo

- Dados  $P_0 [1 \ 1]$ ,  $P_1 [2 \ 3]$ ,  $P_2 [4 \ 3]$ ,  $P_3 [3 \ 1]$ , determinar 7 pontos na curva de Bézier, usando as Eqs. (\*) para  $n = 3$ :

$$\mathbf{P}(t) = \sum_{i=0,3} B_i^3(t) \mathbf{P}_i, \quad (*)$$

$$B_i^3(t) = C_3^i (t-1)^{3-i} t^i, \quad C_3^i = 3! / i!(3-i)!$$

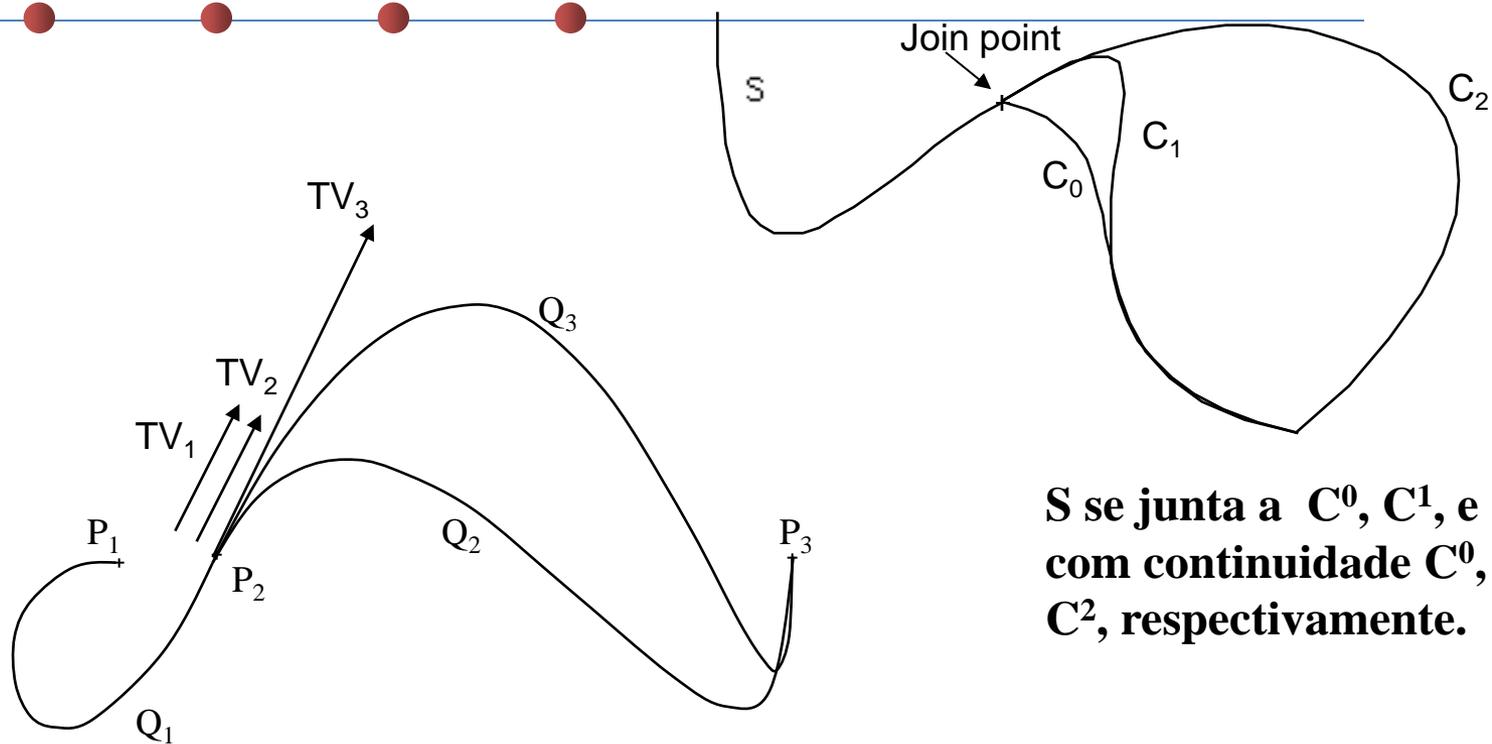
# Curvas de Bézier

- Curvas complexas podem ser obtidas 'concatenando' várias curvas de grau baixo: aproximação por partes
  - Continuidade de ordem 0: junção das curvas (fácil:  $P_n = P'_0$ )
  - Continuidade de ordem 1: tangentes às curvas no ponto de junção são coincidentes:  $P_{n-1}, P_n = P'_0, P'_1$  devem ser colineares

# Continuidade

- Nos pontos de junção das curvas
  - Continuidade geométrica  $G^0$  : dois segmentos de curva se juntam
  - Continuidade geométrica  $G^1$  : os vetores tangentes aos dois segmentos no ponto de junção têm a mesma direção
  - Continuidade paramétrica  $C^1$ : os vetores tangentes aos dois segmentos no ponto de junção têm a mesma direção e mesma magnitude
    - ( $C^1 \Rightarrow G^1$  a menos que o vetor tangente seja =  $[0, 0, 0]$ )
  - Continuidade paramétrica  $C^n$ : direção e magnitude dos vetores tangentes até a n-ésima derivada são iguais no ponto de junção

# Exemplos de Junção



**S se junta a  $C^0$ ,  $C^1$ , e  $C^2$  com continuidade  $C^0$ ,  $C^1$ , e  $C^2$ , respectivamente.**

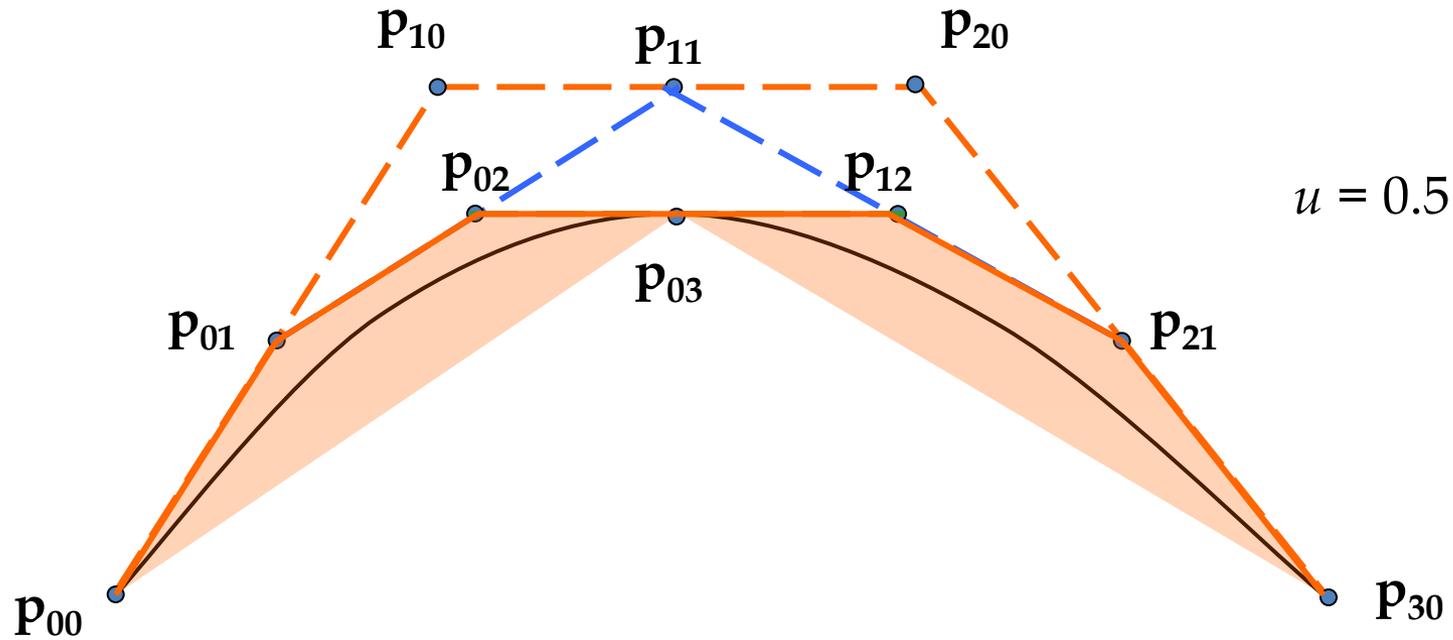
**$Q_1$  e  $Q_2$  têm continuidade  $C^1$  (tangentes  $TV_1$  e  $TV_2$  são iguais).  $Q_1$  e  $Q_3$  têm apenas continuidade  $G^1$ .**

# Desenhando Curvas Bézier

- Curva normalmente é aproximada por uma linha poligonal
- Pontos podem ser obtidos avaliando a curva em  $u$   
 $= u_1, u_2 \dots u_k$ 
  - Avaliar os polinômios de Bernstein
  - Usar o algoritmo recursivo de De Casteljau
- Quantos pontos?
  - Mais pontos em regiões de alta curvatura
- Idéia: subdividir recursivamente a curva em trechos até que cada trecho seja aproximadamente “reto”

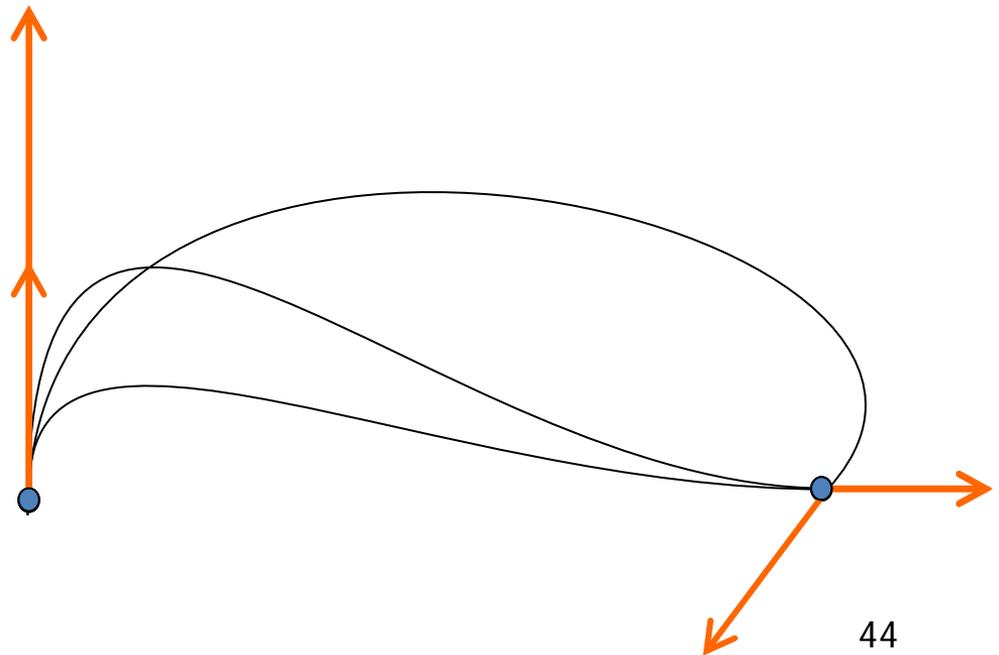
# Subdivisão de Curvas Bézier

- Como saber se trecho da curva é "reto"?
  - Encontrar o polígono de controle do trecho
  - Parar se vértices do polígono forem aproximadamente colineares



# Curvas de Hermite

- Ao invés de modelar a curva a partir de um polígono de controle (Bézier), especifica-se pontos de controle e vetores tangentes nesses pontos
- Vantagem: é fácil emendar várias curvas bastando especificar tangentes iguais nos pontos de emenda
- Exemplos (cúbicas):



# Curvas de Hermite

- No caso de cúbicas, temos o ponto inicial e final além dos vetores tangentes

$$\mathbf{p}(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \mathbf{M}_H \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}'(0) \\ \mathbf{p}'(1) \end{bmatrix}$$

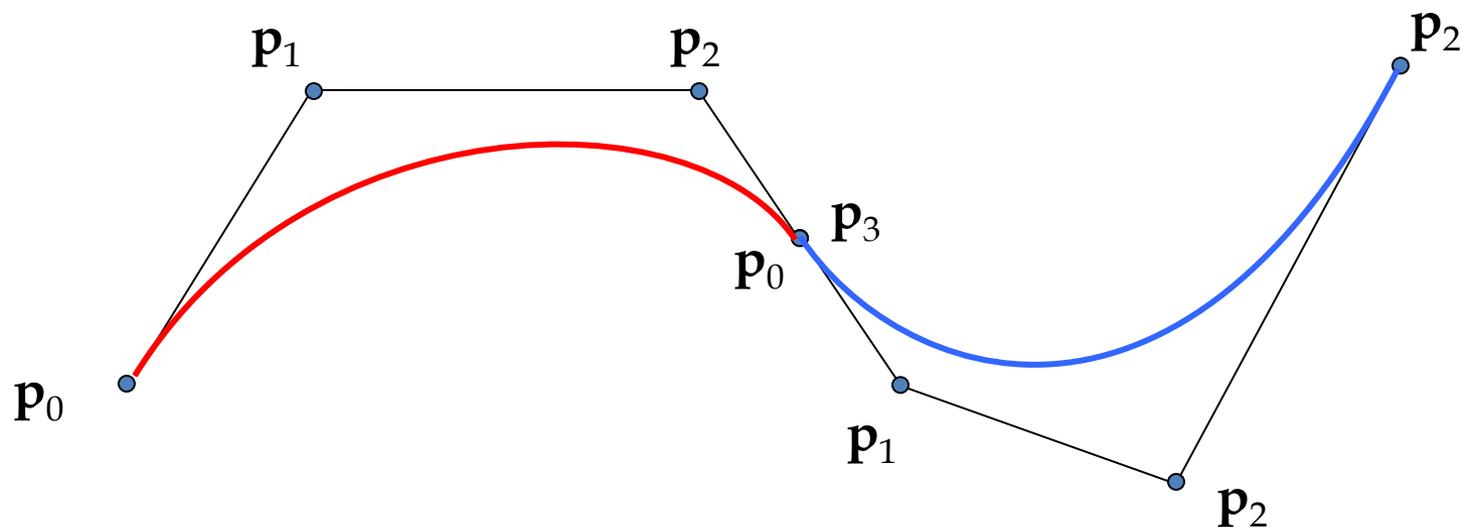
$$\text{onde } \mathbf{M}_H = \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

# Curvas Longas

- Curvas Bézier definidas por  $k$  pontos de controle tem grau  $k - 1$
- Curvas de grau alto são difíceis de desenhar
  - Complexas
  - Sujeitas a erros de precisão
- É desejável que pontos de controle tenham efeito *local*
  - Em curvas Bézier, todos os pontos de controle têm efeito *global*
- Solução:
  - Emendar curvas polinomiais de grau baixo
  - Relaxar condições de continuidade

# Emendando Curvas Bézier

- Continuidade  $C^0$ : Último ponto da primeira = primeiro ponto da segunda
- Continuidade  $C^1$ :  $C^0$  e segmento  $\mathbf{p}_2\mathbf{p}_3$  da primeira com mesma direção e comprimento que o segmento  $\mathbf{p}_0\mathbf{p}_1$  da segunda
- Continuidade  $C^2$ :  $C^1$  e + restrições sobre pontos  $\mathbf{p}_1$  da primeira e  $\mathbf{p}_2$  da segunda



# Superfícies de Bézier

- Analogamente, pode-se definir superfícies que aproximam um conjunto de pontos no espaço
- A formulação matemática de superfícies de Bézier é dada por

$$\mathbf{S}(u,v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) \mathbf{p}_{ij}, 0 \leq u, v \leq 1$$

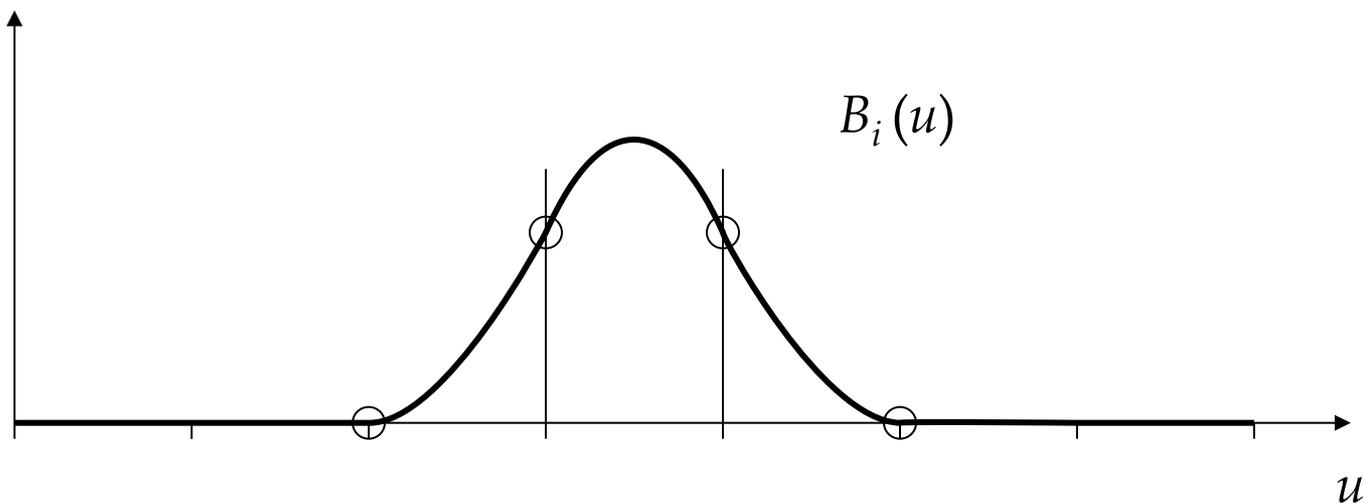
- sendo que  $(n+1)(m+1)$  pontos de controle  $\mathbf{P}_{ij}$  definem um poliedro de controle da superfície
- Ver <http://www.ibiblio.org/e-notes/Splines/Inter.htm>

# Splines

- A base de Bézier não é própria para a modelagem de curvas longas
  - Bézier única: suporte não local
  - Trechos emendados: restrições não são naturais
- Base alternativa: B-Splines
  - Nome vem de um instrumento usado por desenhistas
  - Modelagem por polígonos de controle sem restrições adicionais e com suporte local
    - Alteração de um vértice afeta curva apenas na vizinhança
  - Existem muitos tipos de Splines, mas vamos nos concentrar em B-splines uniformes
    - Uma B-spline uniforme de grau  $d$  tem continuidade  $C^{d-1}$

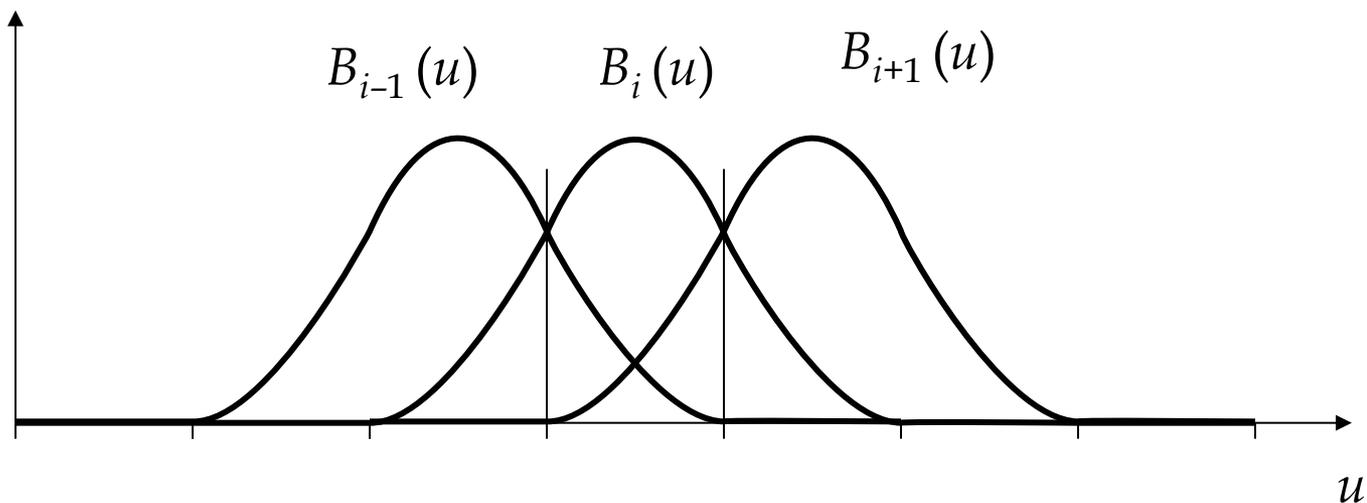
# Curvas B-Spline

- Funções de base são não nulas apenas em um intervalo no espaço do parâmetro
  - Como é impossível obter isso com apenas 1 polinomial, cada função de base é composta da emenda de funções polinomiais
  - Por exemplo, uma função de base de uma B-spline quadrática tem 3 trechos (não nulos) emendados com continuidade  $C^1$



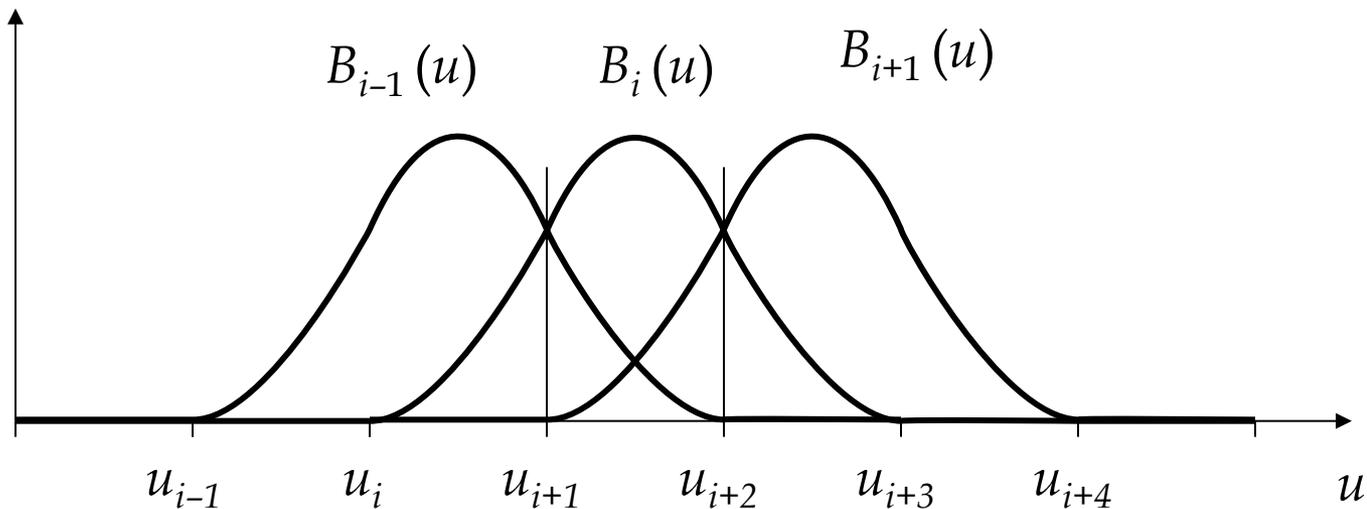
# Curvas B-Spline

- Todas as funções de base têm a mesma forma, mas são deslocadas entre si em intervalos no espaço de parâmetros
- Num determinado intervalo, apenas um pequeno número de funções de base são não-nulas
  - Numa B-spline quadrática, cada intervalo é influenciado por 3 funções de base



# Curvas B-Spline

- Os valores  $u_i$  do espaço de parâmetro que delimitam os intervalos são chamados de *nós*
- Podemos pensar em intervalos regulares por enquanto (B-Splines uniformes) isto é,  $u_i = 1$



# Funções da Base B-Spline

- Queremos exprimir curvas como pontos mesclados por intermédio de funções da base B-Spline

$$\mathbf{p}(u) = \sum_{i=0}^m B_{i,d}(u) \mathbf{p}_i$$

onde  $m$  é o número de pontos do polígono de controle e  $d$  é o grau da B-spline que se quer usar

- Para derivar as funções da base B-spline pode-se resolver um sistema de equações
  - Para B-splines cúbicas, requiere-se continuidade  $C^2$  nos nós, a propriedade do fecho convexo, etc
- Uma maneira mais natural é utilizar a recorrência de Cox-de Boor que exprime as funções da base B-Spline de grau  $k$  como uma interpolação linear das funções de grau  $k-1$

# Recorrência Cox-de Boor

$$B_{k,0}(u) = \begin{cases} 1 & \text{para } u_k \leq u < u_{k+1}, \\ 0 & \text{caso contrário.} \end{cases}$$

$$B_{k,d}(u) = \frac{u - u_k}{u_{k+d} - u_k} B_{k,d-1} + \frac{u_{k+d+1} - u}{u_{k+d+1} - u_{k+1}} B_{k+1,d-1}$$

# Recorrência Cox-de Boor

$$B_{k,0}(u) = \begin{cases} 1 & \text{para } u_k \leq u < u_{k+1}, \\ 0 & \text{caso contrário.} \end{cases}$$

$$\mathbf{p}(u) = \sum_{i=0}^m B_{i,d}(u) \mathbf{p}_i$$

$$B_{k,d}(u) = \frac{u - u_k}{u_{k+d} - u_k} B_{k,d-1} + \frac{u_{k+d+1} - u}{u_{k+d+1} - u_{k+1}} B_{k+1,d-1}$$

$$\mathbf{p}(u_{i+2} \leq u < u_{i+4})$$



$\mathbf{p}_{i+3}$

$$\mathbf{p}(u_i \leq u < u_{i+1})$$

$\mathbf{p}_i$  

$$\mathbf{p}(u_{i+2} \leq u < u_{i+3})$$



$\mathbf{p}_{i+2}$

$$\mathbf{p}(u_{i+1} \leq u < u_{i+2})$$



$\mathbf{p}_{i+1}$

$d = 0$

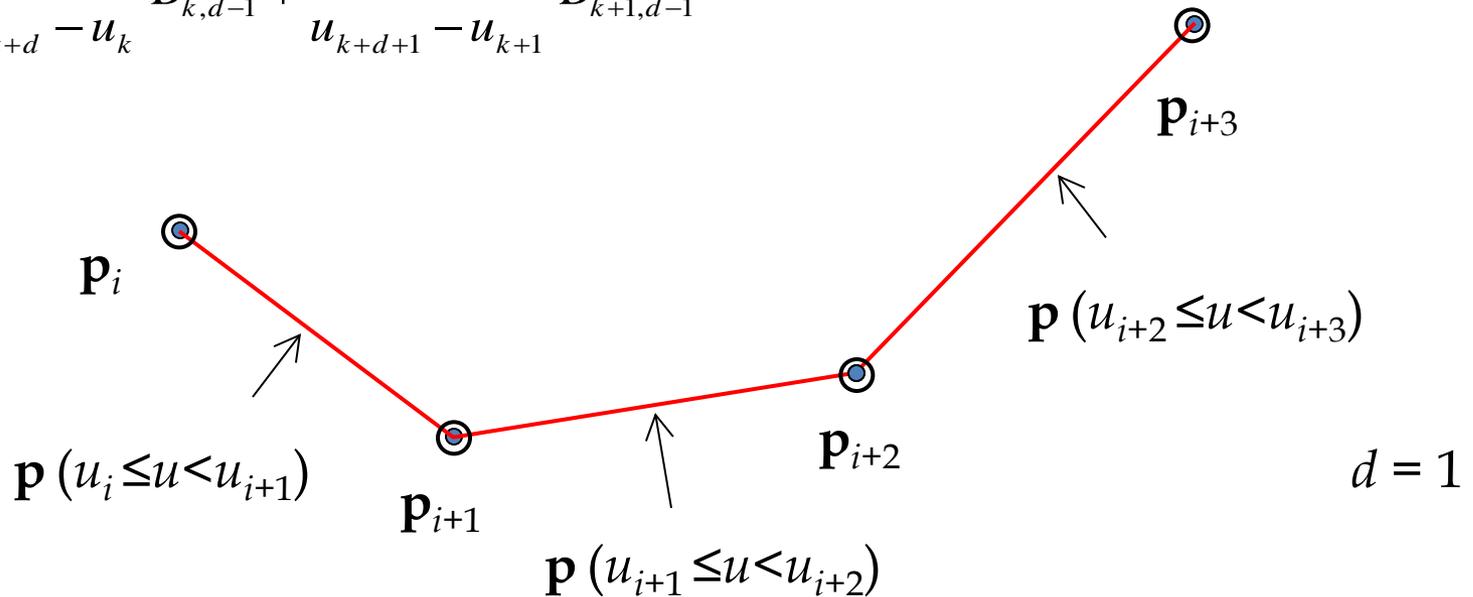
(assumir que para  $u = u_i$   
Spline de grau 0 passa por  $\mathbf{p}_i$ )

# Recorrência Cox-de Boor

$$B_{k,0}(u) = \begin{cases} 1 & \text{para } u_k \leq u < u_{k+1}, \\ 0 & \text{caso contrário.} \end{cases}$$

$$\mathbf{p}(u) = \sum_{i=0}^m B_{i,d}(u) \mathbf{p}_i$$

$$B_{k,d}(u) = \frac{u - u_k}{u_{k+d} - u_k} B_{k,d-1} + \frac{u_{k+d+1} - u}{u_{k+d+1} - u_{k+1}} B_{k+1,d-1}$$

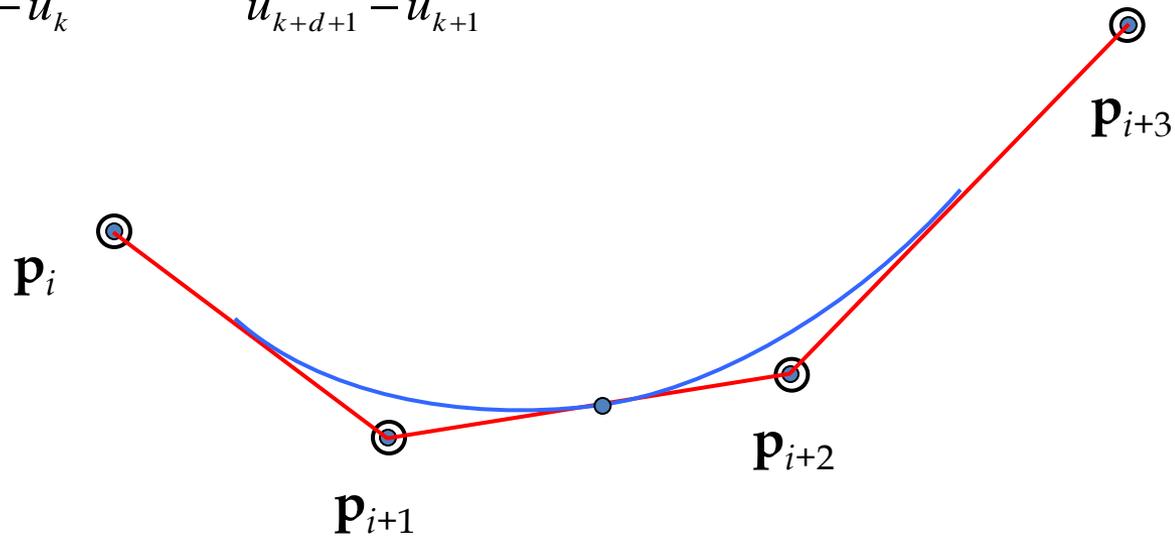


# Recorrência Cox-de Boor

$$B_{k,0}(u) = \begin{cases} 1 & \text{para } u_k \leq u < u_{k+1}, \\ 0 & \text{caso contrário.} \end{cases}$$

$$B_{k,d}(u) = \frac{u - u_k}{u_{k+d} - u_k} B_{k,d-1} + \frac{u_{k+d+1} - u}{u_{k+d+1} - u_{k+1}} B_{k+1,d-1}$$

$$\mathbf{p}(u) = \sum_{i=0}^m B_{i,d}(u) \mathbf{p}_i$$

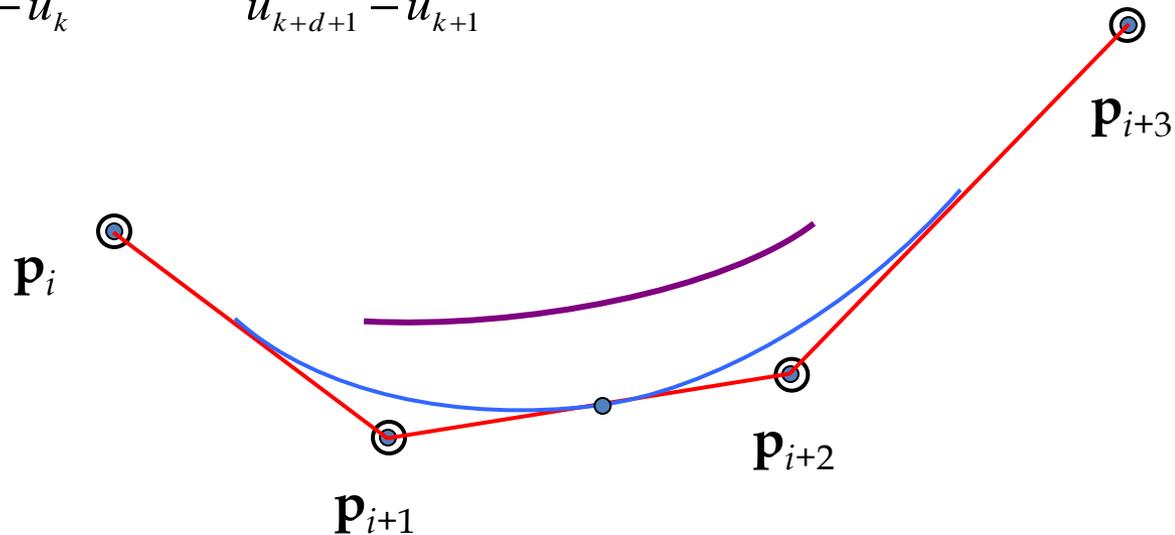


# Recorrência Cox-de Boor

$$B_{k,0}(u) = \begin{cases} 1 & \text{para } u_k \leq u < u_{k+1}, \\ 0 & \text{caso contrário.} \end{cases}$$

$$B_{k,d}(u) = \frac{u - u_k}{u_{k+d} - u_k} B_{k,d-1} + \frac{u_{k+d+1} - u}{u_{k+d+1} - u_{k+1}} B_{k+1,d-1}$$

$$\mathbf{p}(u) = \sum_{i=0}^m B_{i,d}(u) \mathbf{p}_i$$



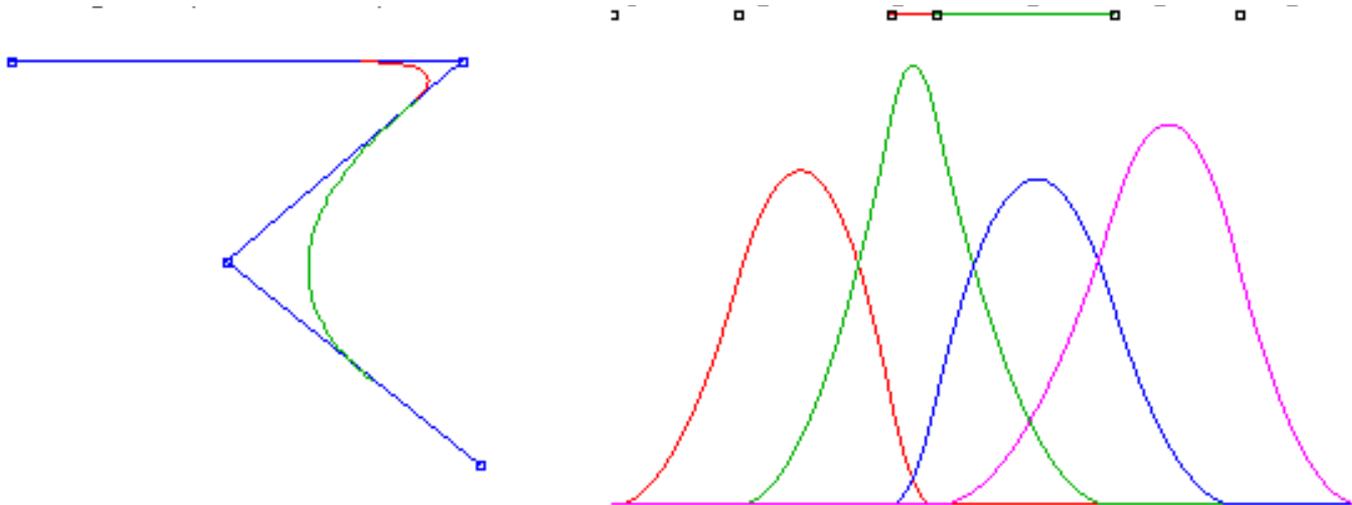
$d = 3$

# Propriedades das B-Splines

- Dados  $n+1$  pontos  $(\mathbf{p}_0 \dots \mathbf{p}_n)$ , é composta de  $(n-d+1)$  curvas Bézier de grau  $d$  emendadas com continuidade  $d-1$  nos  $n+d+1$  nós  $u_0, u_1, \dots, u_{n+d+1}$
- Cada ponto da curva é afetado por  $d+1$  pontos de controle
- Cada ponto de controle afeta  $d+1$  segmentos
- Curva restrita ao fecho convexo do polígono de controle
- Invariância sob transformações afim

# Efeito dos Nós

- Os intervalos entre nós influenciam a importância dos pontos de controle
  - Exemplo: B-spline Quádrica



# Inserindo Nós

- Observamos que as B-splines uniformes em geral não passam pelos pontos de controle
- Entretanto, pode-se replicar nós para fazer a curva se aproximar dos pontos de controle
  - Para interpolar o primeiro ponto usando uma B-Spline cúbica, fazemos  $u_0 = u_1 = u_2 = u_3$
  - Para obter uma B-spline cúbica passando por 4 pontos pode-se usar o vetor de nós: 0, 0, 0, 0, 1, 1, 1, 1
  - De fato, com este vetor de nós, tem-se uma Bézier cúbica

# Vetores de Nós - Exemplos

- Uniforme:  $[0, 1, 2, 3, 4, 5]$
- Uniforme aberto:  $[0, 0, 0, 1, 2, 3, 4, 4, 4]$   
( $k=3, n=5$ )
- Não-uniforme:  $[0, 2, 3, 6, 9]$ 
  - Ver <http://www.ibiblio.org/e-notes/Splines/Basis.htm>
  - Ver <http://www.ibiblio.org/e-notes/Splines/None.htm>

# Curvas Racionais

- Funções são razões

- Avaliados em coordenadas homogêneas:

$$[x(t), y(t), z(t), w(t)] \rightarrow \left[ \frac{x(t)}{w(t)}, \frac{y(t)}{w(t)}, \frac{z(t)}{w(t)} \right]$$

- NURBS (Non-Uniform Rational B-Splines):  $x(t)$ ,  $y(t)$ ,  $z(t)$  e  $w(t)$  são B-splines não uniformes

- Vantagens:

- Invariantes sob transformações perspectivas e portanto podem ser avaliadas no espaço da imagem
- Podem representar perfeitamente seções cônicas tais como círculos, elipses, etc

# B-splines Racionais não-uniformes (NURBS)

- Curva B-spline é uma soma ponderada dos seus pontos de controle

$$\mathbf{P}(t) = \sum_{i=0,n} B_{i,k}(t) \mathbf{p}_i, \quad t_{k-1} \leq t \leq t_{n+1} \quad (*)$$

- os pesos  $N_{i,k}$  têm a propriedade  $\sum_{i=0,n} N_{i,k}(t) = 1$

- Como os pesos dependem apenas do vetor de nós, é útil associar a cada ponto de controle um peso extra  $w_i$

$$\mathbf{P}(t) = \sum_{i=0,n} w_i B_{i,k}(t) \mathbf{p}_i / \sum_{i=0,n} w_i B_{i,k}(t) \quad (**)$$

# B-splines Racionais não-uniformes (NURBS)

- Aumentar o peso  $w_i$  aumenta a influência do  $i$ -ésimo ponto de controle, atraindo a curva para esse ponto
- Denominador em (\*\*) normaliza os pesos: se  $w_i = \text{const}$  para todo  $i$  obtém-se a Eq. (\*)
- Os pesos  $w_i N_{i,k}$  também satisfazem a condição da “partição da unidade”
- Ver <http://www.ibiblio.org/e-notes/Splines/Basis.htm>

# Curvas NURBS

- Representação exata para seções cônicas (elipses, parábolas, hipérbolas)
  - Representação única para todos os tipos de curvas!
- Invariantes sob transformações de projeção perspectiva
  - Transformações podem ser aplicadas aos pontos de controle!

# Superfícies NURBS

- Analogamente, uma superfície NURBS  $\mathbf{S}(u,v)$  é construída como o produto tensorial de duas curvas NURBS:

$$\mathbf{S}(u,v) = \frac{\sum_{i=0,nu} \sum_{j=0,nv} N_i^{Ku}(u) N_j^{Kv}(v) w_{ij} \mathbf{P}_{ij}}{\sum_{i=0,nu} \sum_{j=0,nv} N_i^{Ku}(u) N_j^{Kv}(v) w_{ij}}$$

- Ver

<http://www.ibiblio.org/enotes/Splines/Intro.htm>

<http://libnurbs.sourceforge.net/index.shtml>

# Bibliografia adicional

- Cap. 3 do livro de Azevedo e Conci
- <http://www.cs.princeton.edu/~min/cs426/jar/bezier.html>
- <http://www.cse.unsw.edu.au/~lambert/splines/>